A simple strategy to prune neural networks with an application to economic time series

by

Johan F. Kaashoek and Herman K. van Dijk

Econometric Institute and Tinbergen Institute
Erasmus University Rotterdam
P.O.Box 1738, 3000 DR Rotterdam
The Netherlands

Abstract

A major problem in applying neural networks is specifying the size of the network. Even for moderately sized networks the number of parameters may become large compared to the number of data. In this paper network performance is examined while reducing the size of the network through the use of multiple correlation coefficients and graphical analysis of network output per hidden layer cell and input layer cell.

Contents

1 Introduction

The fame to handle complex data, may have contributed considerably to the diffusion and implementation of neural network models, also in economics and econometrics; see e.g. Hecht-Nielsen [4], Hertz, Krogh & Palmer [5], Gallant & White [3] and White [13].

In this paper only one type of neural network is considered: a feed-forward 3-layer network with an input layer of I cells (nodes), a hidden layer with H cells (nodes) and an output layer with O cells; the network will be denoted as $nn(I, H, O)$. The functional form of this network is

$$y = C\, G(A\, x + b) + d, \tag{1}$$

with A an $I \times H$ matrix, b a H vector, C a $H \times O$ matrix and d a O vector; G is a multi-valued non-linear (activation) function.
It is assumed that this neural network is an approximation of the data generating process:

$$x_t = F(x_{t-1}, \cdots, x_{t-N}) + \epsilon_{t-1}, \tag{2}$$

where $x_t \in \mathbb{R}$, F the data generating function and ϵ_{t-1} represents an unknown noise term. Hence the size O (dimension) of the output layer is a priori given and equal to 1. An upperbound N on the size of the input layer is given by nonlinear data analysis (e.g. embedding dimension, see Takens [11]) but the size H of the hidden layer is unknown and has to be determined.

The flexibility of a neural network makes that overfitting, i.e. fitting the noise process, and consequently bad prediction behaviour can easily occur, see Bishop [1].

A two fold procedure is applied in reducing the network size. The starting-point is what is called by Theil (see Theil [12]): the incremental contribution of variables. That is, how much more of the variance of the dependent variable y is explained by inclusion of e.g. the hth explanory variable given that all other variables are used. In Theil (o.c) the incremental contribution is measured in terms of the multiple correlation coefficients. A variable with a low incremental contribution will be a candidate to be excluded from the model. Secondly, graphical analysis of network output with exclusion of network nodes is used as justification for in- or exclusion of variables with low incremental contribution. As a node pruning method, this approach is similar to the one proposed by Mozer and Smolensky [7]. The approach of Theil however has the advantage that the quantities used are based on the outcome of one optimization procedure with all variables included.

The paper is organized as follows. In the first part the term incremental contribution and multiple correlation coefficients are briefly explained. In the second part pruning of a network based on incremental contribution of variables is introduced.
In the third part the procedure is applied to two examples; in the first one the data are generated by a completely deterministic process while the second example concerns actual economic data: the logarithm of Yen-US Dollar real exchange rates.

2 Incremental contribution of variables

Consider a linear model:

$$y = X\,A + b + \epsilon, \tag{3}$$

with $y \in \mathbb{R}$, A a vector of unknown parameters with length H, b a constant term and ϵ some stochastic process. The multiple correlation coefficient R associated with the least squares regression of y on $(X\;1)$ is given as:

$$R^2 = 1 - \frac{e'e}{(y-\overline{y})'(y-\overline{y})} \tag{4}$$

where e represents the residuals of the least squares regression and $\overline{y}$ the mean of y. Now regress again with the hth variable not included; the residuals obtained in this way are denoted as e_h and the multiple correlation coefficient R_h is defined by

$$R_h^2 = 1 - \frac{e_h'e_h}{(y-\overline{y})'(y-\overline{y})}. \tag{5}$$

The quantity $R^2 - R_h^2$ is called the incremental contribution of the variable h in explaining the variance of y, see Theil [12]. The two regressions will , in general, not give equal estimates of A and b; only if the hth column of X is orthogonal to all other columns the two regression give equal estimates. If all columns of X are pair wise orthogonal then the following relation holds:

$$\sum_{h=1}^{H}(R^2 - R_h^2) = R^2. \tag{6}$$

Now consider the regression with all variables included; this gives the residual e and the multiple correlation coefficient R. By putting the hth variable to zero, one gets residuals e_h and multiple correlation coefficient R_h as in equation (5). The contribution of the variable with index h in explaining the variance of y is measured as $R^2 - R_h^2$.

$$R^2 - R_h^2 = \frac{e_h'e_h - e'e}{(y-\overline{y})'(y-\overline{y})} \tag{7}$$

The discrepancy between R^2 and the sum over all incremental contributions is called the multicollinearity effect. The inclusion or exclusion of variables in the final model is now based on their incremental contributions $R^2 - R_h^2$. Variables with low contributions are the first candidates to be excluded.

By means of the multiple correlation coefficients R_h the partial correlation coefficient r_h is defined as:

$$r_h = \sqrt{\frac{R^2 - R_h^2}{1 - R_h^2}} \tag{8}$$

Note that r_h is the correlation coefficient of the regression model

$$e_h = X_h A_h + e \tag{9}$$

where X_h is hth column of X and A_h is $h$the parameter. So r_h measures how much of the unexplained variance in y with inclusion of all variables except hth variable, is explained by the hth variable. The disadvantage of using r_h as a measure for inclusion or exclusion is that with $R^2 \sim 1$, $r_h \sim 1$ independent of the value of R_h^2. So Theil is followed (see Theil [12]) and the incremental contributions $R^2 - R_h^2$, see equation (7) are used as a measure.

3 Network pruning using multiple correlation coefficients

The functional form of the network used is given as:

$$\begin{aligned} y =& c\,G(A\,x + b) + d, \\ A =& [a_{hi}],\ \ H \times I \text{ matrix of connection weights}, \\ b =& (b_1, \cdots, b_H)',\ \ \text{vector of internal thresholds}, \\ c =& (c_1, \cdots, c_H)',\ \ \text{vector of connection weights}, \\ d =&\ \text{output constant}. \\ G(x_1, \cdots, x_H) =& (g(x_1), \cdots, g(x_H))',\ \ G : \mathbb{R}^H \to \mathbb{R}^H, \end{aligned} \tag{10}$$

Denoting the output of layer cells with $h = (h_1, \cdots, h_H)'$ then the network can be written as

$$y = c\,h + d \tag{11}$$

$$h_k = g((A\,x)_k + b_k), k = 1, \cdots, H. \tag{12}$$

where g is an activation function; e.g. $g(x) = \frac{1}{1+e^{-x}}$.

Suppose some time series $\{y(t), x(t)\}$ are fitted to the network given by (10). After some optimization procedure, for each hidden layer cell the incremental contribution to R^2 can be calculated (given A, b and d). Let $\hat{y}$ be the network output with inclusion of all cells, $\hat{y}_h$ network output with hidden layer cell excluded, then the incremental contribution of hidden cell h is again:

$$\begin{aligned} R^2 - R_h^2 =& \frac{e'e - e_h' e_h}{(y - \overline{y})'(y - \overline{y})} \\ e =& y - \hat{y}, \\ e_h =& y - \hat{y}_h. \end{aligned} \tag{13}$$

If the value in (13) is low for some h compared to all other values, then this cell is a candidate for exclusion from the network. This decision can be confirmed by some graphical analysis.
The graphs of $\{t, \hat{y}_h(t)\}$ compared to the graph of $\{t, y(t)\}$ will give evidence of the contribution of each hidden cell h in explaining the variance of $y(t)$. The best way for comparison is to adjust all series $\{y_t\}$ and $\{\hat{y}_h(t)\}, h = 1, \cdots, H$ for mean. Which the network configuration given by equation (10) this is no problem as the constant d can always take care of mean differences.

If some cell h is excluded from the network and corresponding parameters are deleted, the optimization procedure is prolonged with all other parameters unchanged except the parameter d which is adjusted for mean differences between original data $\{y_t\}$ and network output $\{\hat{y}_h(t)\}$[1].

The same procedure can be applied to reduce the number of input layer cells. In general, economic considerations will determine which variables are to be included; in the case of a time series model as in (2) the number of input cells equals the number of lagged variables used. For instance in the case of pure deterministic time series, the number of input cells is bounded from above by the embedding dimension of the observed series $\{x_t\}$, see e.g. Takens [11]. However, even if this embedding dimension can be determined, it is only an upperbound and further econometric analysis can be applied which can lead to a reduction of the number of input cells.

4 Two examples

In this section two examples are given which illustrate the pruning method explained above.

4.1 Deterministic data

The first example has to do with data generated by the following deterministic process:

$$\begin{aligned} x_t &= 0.95\, x_{t-1} + (\epsilon_{t-1} - 0.5) \\ \epsilon_t &= 4\, \epsilon_{t-1}(1 - \epsilon_{t-1}) \end{aligned} \tag{14}$$

So, the series x_t is an AR-process with disturbances $\epsilon_{t-1} - 0.5$ generated by the logistic map; as is well-known, the series ϵ_t is chaotic.
The observed data are the series $\{x_t\}$ only. The number of data used is 200. The data set is called by AR_{CH}.

[1] Apart from consistency in the definition of multiple correlation coefficients, which are defined in deviation of means, the inclusion of the constant d in the network definition is motivated by the possibility to adjust easily network output $\{\hat{y}_h(t)\}$ for differences in mean.

The process (14) is to be reconstructed using a neural network. From non-linear data analysis it follows that an upperbound on the embedding (number of lags) is given by $2N+1$ where N is the dimension of the original data generating process; in our case $N=2$. Hence our starting point is the unknown relation:

$$x_t = F(x_{t-1}, \cdots, x_{t-5}) \tag{15}$$

and F is to be approximated by a neural network with input dimension I equal to 5.

The dimension of the hidden layer is taken to be 5; hence $H = 5$.

As "learning" method a non-linear optimization procedure is applied; to be more specific, a variable metric method known as Davidon-Fletcher-Powell, see Press e.a.,[8] with object function the sum of squared residuals. The results of the optimization can be found in table 1. The first column of the table gives the cells which are excluded: jH means cell j of the hidden layer; jI similar but now for the input layer. Apart from the multiple correlation coefficients and their incremental contribution, the so-called information criterion, SIC, are reported in the tables.

$$SIC = \ln(MSSR) + \frac{n_p}{2T} ln(T), \tag{16}$$

where n_p is the number of parameters, T is the length of data set and $MSSR$ is the mean sum of squared residuals; see Schwartz [10]. The value of SIC measures abundance of parameters.

Table 1: Multiple correlation coefficient and incremental contribution

Network $(5,5,1)$ Data: AR_{CH}

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9999	.	−6.6921
H1	0.0060	0.9939	0.3145
H2	0.9981	0.0018	−3.0753
H3	0.9570	0.0429	−0.6824
H4	0.3712	0.6287	0.1107
H5	0.9945	0.0054	−1.7397
I1	0.4881	0.5119	1.0926
I2	0.7318	0.2681	1.0537
I3	0.9826	0.0173	−2.6704
I4	0.9998	0.0001	−5.8035
I5	0.9998	0.0001	−5.4982

From table 1 it is obvious that hidden layer cells 2, 3 and 5 can be excluded. No graphs are supplied because the values of the incremental contribution (and also the SIC-values) don't need further illustration.
So hidden cells 2, 3 and 5 are excluded from the net; the constant term d is adjusted for mean difference between actual data and network output and a further optimization run is applied to the reduced network. The results are reported in table 2.

Table 2: Multiple correlation coefficient and incremental contribution

Network $(5,2,1)$ Data: AR_{CH}

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9999	.	−6.9101
H1	0.0460	0.9530	2.5508
H2	0.1176	0.8823	3.2589
I1	0.7881	0.2451	2.2912
I2	0.8318	0.1538	1.8880
I3	0.9998	0.0001	−6.0045
I4	0.9999	0.0000	−6.5175
I5	0.9999	0.0000	−6.6974

As the contribution of each hidden layer cell is almost equal, no further reduction in hidden layer cells is applied. However, input cells 3, 4 and 5 are obvious candidates for exclusion. So the network is reduced to 2 input cells and 2 hidden layer cells. After optimization the results are those reported in table 3.

Table 3: Multiple correlation coefficient and incremental contribution

Network $(2,2,1)$ Data: AR_{CH}

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9999	.	−7.2000
H1	0.0504	0.9495	2.5543
H2	0.1200	0.8799	3.1321
I1	0.7572	0.2427	2.1733
I2	0.8438	0.1561	1.8609

No attempts of further reduction are applied: the contribution of each cell in a layer is similar.

Two graphs show the performance of the final network $nn(2,2,1)$. The first graph gives the time series of the actual data $\{x_t\}$ and the neural network estimates $\{\hat{x}_t\}$, the one period ahead prediction, see figure (1).

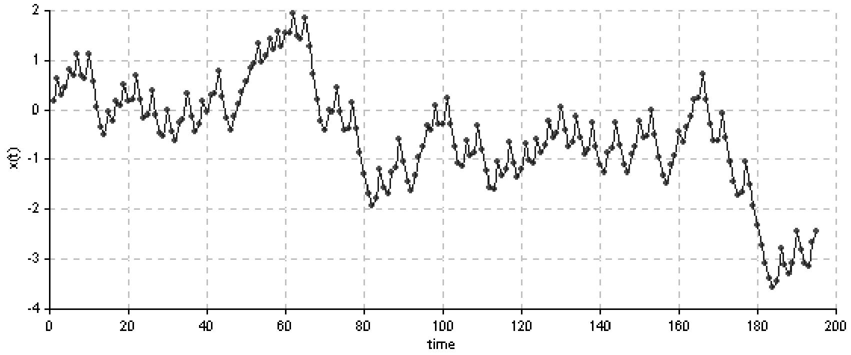

Figure 1: AR_{CH} data (dots) and neural network estimates (continuous curve)

The second graph compares the series $\{x_t\}$ with a series nn_t which is generated as follows: the initial value is given as $nn_t = x_t, t = 1, \cdots 2$, and from that all other values $nn_t, t > 2$ are generated by the neural network. This time series is denoted as "a neural network generated orbit". Since such a "neural network generated orbit" may start at any value (= time index) and can be prolonged for any time period, it shows the prediction capability of the network function over any period at any time; see figure (2).

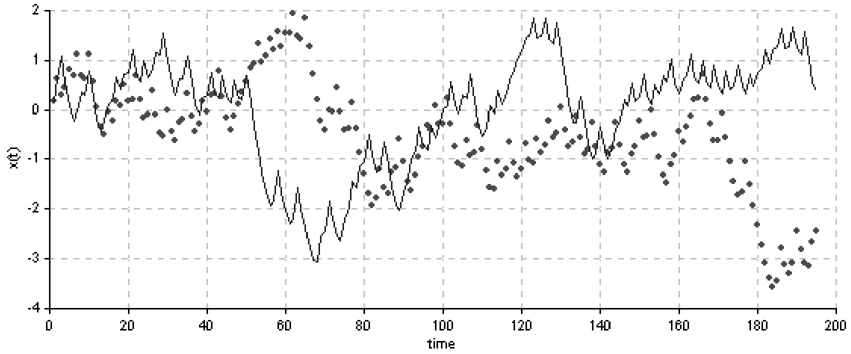

Figure 2: AR_{CH} data (thick dots) and orbit generated by neural network (continuous curve)

The orbit generated by the network function deviates from the actual data after 5 time steps. Note that the "error" $\epsilon_{t-1} - 0.5$ in equation (14) is generated by a structural unstable (and chaotic) system: small deviations in the original function form of ϵ_t can result in considerable and essential deviations in orbits; see e.g. Devaney,[2]. As the neural network approximates the original system, such deviations in prediction are not surprising.

4.2 Economic data

The data used in this section are the logarithm of Yen-US dollar real exchange rates, period December 1972 to June 1988, denoted as $JPUS$; for more details on the data, see Schotman and van Dijk [9] who fitted a linear autoregressive model of order one, $AR1$.

The initial network is rather large: $nn(5,7,1)$. In Kaashoek and van Dijk [6] it is shown that the series $JPUS$ has an embedding dimension of 5 at the most; this means that only variables with a lag of 5 or less needs to be included.

The results of optimization are summarized in table 4.

Table 4: Multiple correlation coefficient and incremental contribution

Network $(5,7,1)$ Data: $JPUS$

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9810	.	−2.9997
H1	0.0075	0.9735	1.7740
H2	0.1722	0.8088	−0.3818
H3	0.9377	0.0433	−1.3314
H4	0.6236	0.3574	−0.2242
H5	0.2578	0.7232	0.0876
H6	0.9698	0.0112	−2.5207
H7	0.0028	0.9781	1.7461
I1	0.0108	0.9702	−0.9798
I2	0.5728	0.4082	−1.0869
I3	0.4727	0.5083	−1.3049
I4	0.4512	0.5298	−0.9664
I5	0.3641	0.6169	−1.2264

From table 4 one can conclude that hidden layer cells 3 and 6 may be excluded. This is further illustrated in figure 3 and figure 4. In this two figures, the neural network output with exclusion of one hidden layer cell each time is compared with actual data $JPUS$. In each figure the neural network output is always adjusted for mean differences.

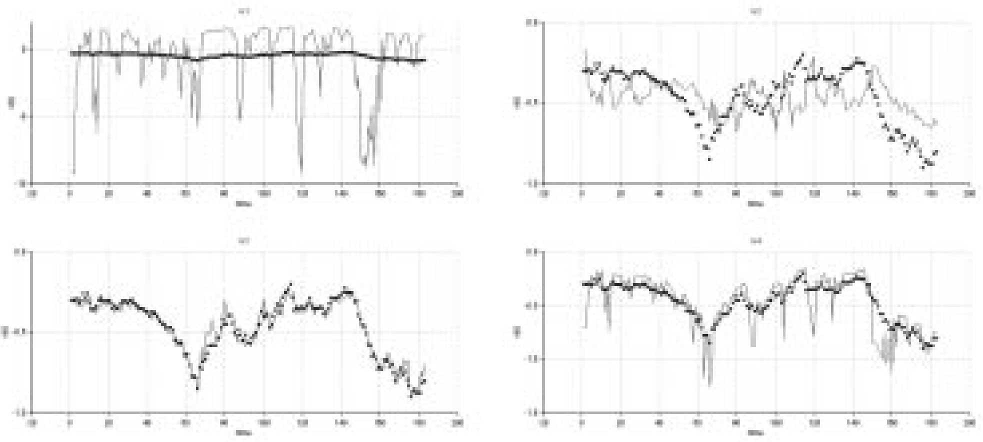

Figure 3: $JPUS$ data (thick dots) and $nn(5,7,1)$ network output without hidden layer cell 1, 2, 3 and 4.

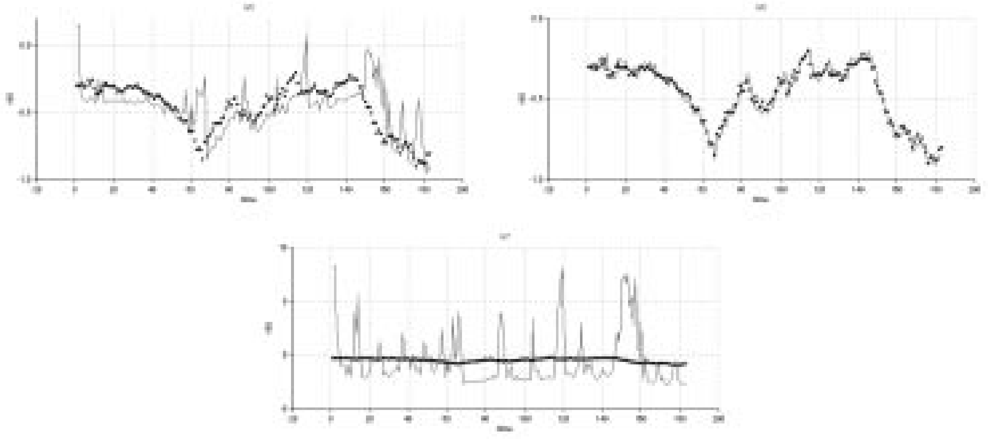

Figure 4: $JPUS$ data (thick dots) and $nn(5,7,1)$ network output without hidden layer cell 5, 6 and 7.

With respect to the input layer cells, according to the results reported in table 4 all input variables has to be included (so far). Again, this is illustrated by a figure: figure 5 shows network output with exclusion of each input layer cell separately. Especially, input layer cell 1, variable x_{t-1} and input layer cell 5, variable x_{t-5}, have in this configuration a large contribution, and can not be excluded.

After exclusion of hidden layer cell 3 and 6, another optimization round is applied. In table 5 the results of the network $nn(5,5,1)$ are summarized.

The results in table 5 are confirmed by examination of the corresponding graphs; in this case only the graphs of network output minus one hidden layer cell are shown, see figure 6.

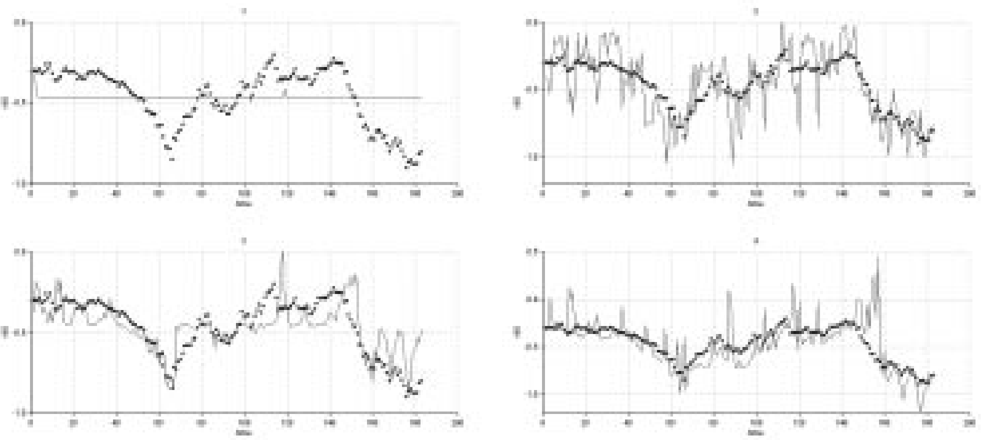

Figure 5: $JPUS$ data (thick dots) and $nn(5,7,1)$ network output without input layer cell 1, 2, 3 and 4.

The only candidates for exclusion based on small incremental correlation contributions, are hidden layer cell 3 and 4. A similar conclusion can be conceived from the graphs of figure 6. Hence the process is continued after exclusion of hidden layer cells 3 and 4; the results of optimization are reported in table 6.
As table 6 gives no evidence for exclusion of hidden layer cells, only the graphs of network output minus one input layer cell are reported; see figure 7.

Figure 7 is conform table 6 in the sense that the incremental contributions of input cells 2, 3, 4 and 5) are low; so a network $nn(2,3,1)$ with only two input cells, the variables x_{t-1} and x_{t-2} should be a proper guess. The results after optimization are reported in table 7; it shows a low contribution of the variable x_{t-2}, input cell 2. Figure 8 show actual data and network estimates. As a performance test of this network, the graphs of network prediction, called network orbit, see page 9, and actual data is shown in figure 9.

The performances of networks $nn(5,3,1)$ and $nn(2,3,1)$ differ not much: based on SIC value the choice would be the network $nn(2,3,1)$. Although the input variable x_{t-2} has a very low contribution in explaining the variance of x_t, no further reduction is applied; in Kaashoek and van Dijk [6] it is shown that the data $JPUS$ have an embedding dimension of at least 2; that is, the data are to be modeled by at least x_{t-1} and x_{t-2}.

Table 5: Multiple correlation coefficient and incremental contribution

Network $(5,5,1)$ Data: $JPUS$

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9745	.	−3.0521
H1	0.0119	0.9626	0.7563
H2	0.1873	0.7872	−0.6145
H3	0.9189	0.0556	−1.1956
H4	0.8718	0.1027	−1.0324
H5	0.0495	0.9250	0.6569
I1	0.0113	0.9632	−1.2504
I2	0.2371	0.7374	−1.0855
I3	0.6999	0.2746	−1.8611
I4	0.7731	0.2014	−1.4533
I5	0.4070	0.5675	−1.2454

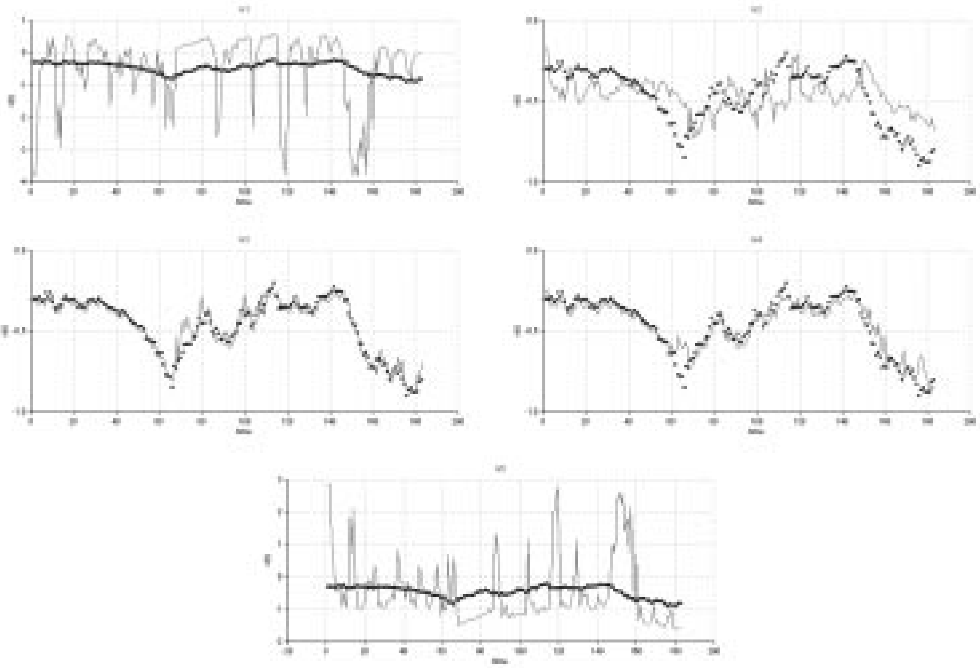

Figure 6: $JPUS$ data (thick dots) and $nn(5,5,1)$ network output without hidden layer cell 1, 2, 3, 4 and 5.

Table 6: Multiple correlation coefficient and incremental contribution

Network $(5,3,1)$ Data: $JPUS$

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9705	.	−3.1786
H1	0.1704	0.8001	1.2493
H2	0.1636	0.8069	−0.4931
H3	0.0199	0.9506	−1.1974
I1	0.1946	0.7759	−0.9046
I2	0.8672	0.1033	−2.1275
I3	0.9048	0.0657	−2.5196
I4	0.8609	0.1096	−2.2962
I5	0.8318	0.1387	−1.1524

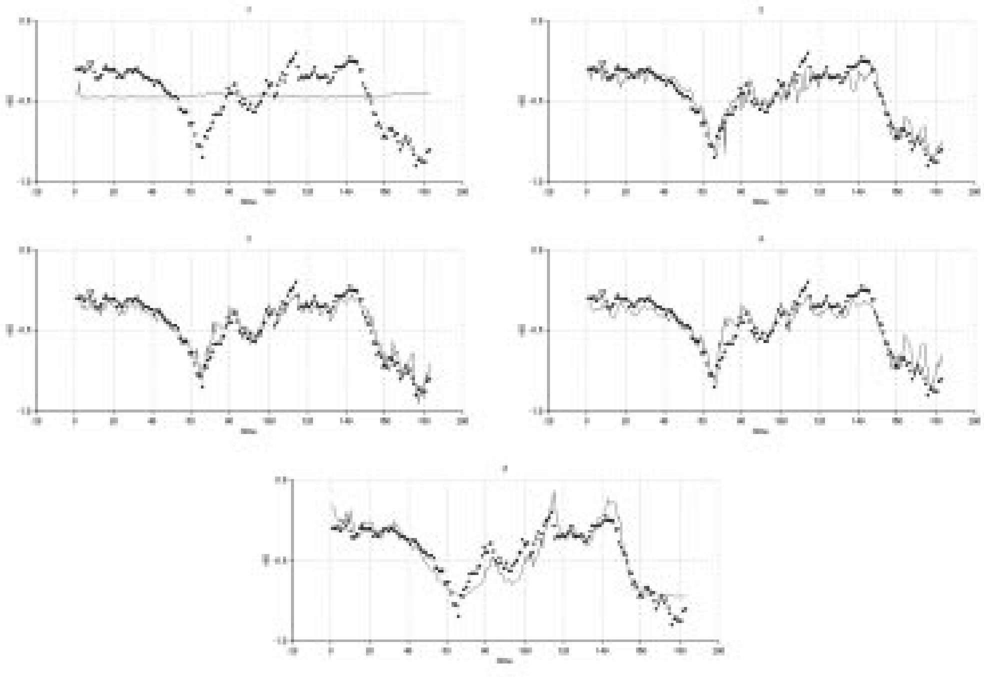

Figure 7: $JPUS$ data (thick dots) and $nn(5,3,1)$ network output without input layer cell 1, 2, 3, 4 and 5.

Table 7: Multiple correlation coefficient and incremental contribution

Network $(2, 3, 1)$	Data: $JPUS$		
Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9673	.	-3.2547
H1	0.8525	0.1148	2.7181
H2	0.3826	0.5847	-0.8744
H3	0.7030	0.2643	1.2156
I1	0.1156	0.8516	2.4523
I2	0.9636	0.0036	-2.5416

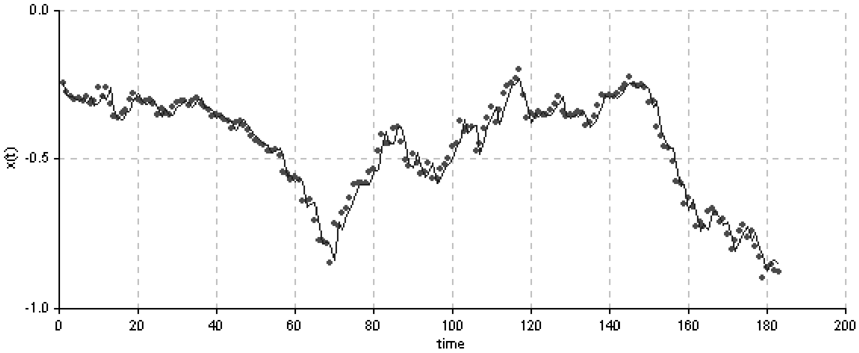

Figure 8: $JPUS$ data (thick dots) and network $nn(2, 3, 1)$ estimates.

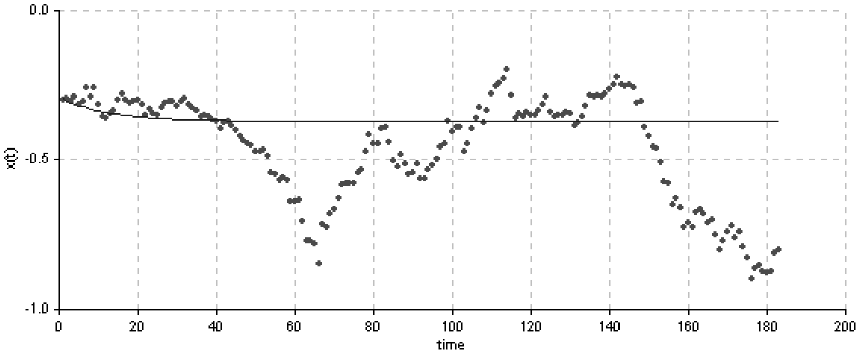

Figure 9: $JPUS$ data (thick dots) and orbit (prediction) generated by $nn(2, 3, 1)$ network.

5 Summary

In this paper a well known procedure in reducing the number of parameters of a linear model is applied as pruning tool for neural networks. The applied neural network configuration where output is linearly connected to hidden layer cells, seems to be appropriate for applying this procedure on reducing the number of hidden layer cells as is shown in the two examples exposed above. Further examination, e.g. F-statistics of partial correlation coefficients (given estimates of parameters not involved) may provide further evidence.

Inclusion of input variables is based on economic analysis and non-linear data analysis; however in most cases one has only an upperbound (and may be a lowerbound) on the number of variables to be included. So reduction of input layer cells may also be possible in this case and, also in this case with nonlinear connections between input and output layer, incremental contribution of correlation coefficients and graphical analysis provide an easy tool to examining neural network performance and reduction of parameters.

References

[1] Bishop, C.M., *Neural Networks for Pattern Recognition*, Clarendon Press Oxford, 1995.

[2] Devaney, R.L., *An Introduction to Chaotic Dynamical Systems*, second ed., Addison-Wesley Publ. Co., Reading, 1989.

[3] Gallant, A.R. & H. White, There exists a neural network that does not make avoidable mistakes, in *Proc. of the International Conference on Neural Networks, San Diego, 1988*, IEEE Press, New York, 1989.

[4] Hecht-Nielsen, R., *Neurocomputing*, Addison-Wesley Publ. Co., Menlo Park, CA, 1990.

[5] Hertz, J., A. Krogh & R.G. Palmer, *Introduction to the theory of neural computation*, Addison-Wesley Publishing Company, Reading Massachusetts, 1991.

[6] Kaashoek, J.F. & H.K. van Dijk, Evaluation and Application of numerical procedures to calculate Lyapunov exponents, *Econometric Reviews*, special issue, Vol. 13, No.1, 1994.

[7] Mozer, M.C. & P. Smolensky, Skeletonization: a technique for trimming the fat from a network via relevance assessment, in D.S. Touretzky (Ed.) *Advances in neural Information Processing Systems*, vol. 1, San Mateo, CA., 1989.

[8] Press, W.H., B.P. Flannery, S.A. Teukolsky & W.T. Vetterling, *Numerical Recipes*, Cambridge University Press, Cambridge, 1988.

[9] Schotman, P. & H.K. van Dijk, On Bayesian routes to unit roots, *Journal of Applied Econometrics*, 1991

[10] Schwartz, G., Estimating the Dimension of a Model, *The Annals of Statistics*, **6**, 1978.

[11] Takens, F., Detecting strange attractors in turbulence, in D.A. Rand and L.S. Young (eds.), *Dynamical systems and turbulence*, Springer-Verlag, Berlin, 1981.

[12] Theil, H., *Principle of Econometrics*, Wiley & Sons, 1971

[13] White, H., Some Asymptotic Results for Learning on Single Hidden Layer Feedforward Network Models, *Journal of the American Statistical Association*,vol.**84**, no.408, 1989.